REMARKS

I. General

Claims 1-20 were pending in the present application. In response to Applicant's previous response, the Final Office Action (mailed February 25, 2008) withdraws the previous grounds of rejection and presents a new ground of rejection, rejecting all of the pending claims 1-20. The new ground of rejection raised in the Final Office Action is:

- Claims 1-20 are rejected under 35 U.S.C. §102(e) as being anticipated by U.S. Patent No. 6,792,454 to Nakano et al. (hereinafter "*Nakano*").

Applicant respectfully traverses the outstanding claim rejections presented in the Final Office Action, and requests reconsideration and withdrawal thereof in light of the remarks presented herein.

II. Rejections Under 35 U.S.C. §102(e) over *Nakano*

Claims 1-20 are rejected under 35 U.S.C. §102(e) as being anticipated by *Nakano*. To anticipate a claim under 35 U.S.C. § 102, a single reference must teach every element of the claim, *see* M.P.E.P. § 2131. Applicant respectfully submits that claims 1-20 are not anticipated by *Nakano* because *Nakano* fails to teach each and every element of the claims, as discussed further below.

Independent Claim 1

Independent claim 1 recites:

> A method for staging file assets on a live server comprising:
> detecting an index page of said server;
> creating a staging folder within a file system of said server, wherein said staging folder does not default to a directory listing of said file system when accessed;
> inserting a randomized string into a name of said file assets to be staged; and
> storing said file assets in said staging folder. (Emphasis added).

Nakano fails to teach at least the above-emphasized elements of claim 1, as discussed further below.

First, *Nakano* does not teach a method for staging file assets "on a live server", but instead teaches a staging area that is implemented on a development server 130 as opposed to its live "website production server" 170, *see* Fig. 1 of *Nakano* and *see* col. 5, lines 5-23 of *Nakano*. As can be seen above, the preamble of claim 1 refers to staging file assets on a live server, and the body of the claim refers back to "said server", thus referring to the live server. The present application clearly describes at, for example, paragraphs 0016-0017 in connection with its description of FIGURES 1-2, that the staging of file assets is performed on the live server. Indeed, FIGURE 1 is described in paragraph 0016 as illustrating a traditional technique similar to that of *Nakano* in which a staging web server 104 (such as *Nakano's* development server 130) is used for containing new and developing web content, while a separate live web server 103 (such as *Nakano's* server 170) is used for hosting the live web content that is accessible to public users 101 and 102. On the other hand, paragraph 0017 describes FIGURE 2 in which a separate staging server is not required, but instead the staging of file asserts occurs on the live web server. *Nakano* fails to teach any such method for staging file assets on a live server, but (like the example of FIGURE 1 of the present application) merely teaches staging file assets that are under development at a separate development server 130. Thus, *Nakano* fails to anticipate claim 1 for at least this reason.

Second, *Nakano* does not teach creating a staging folder within a file system of "said server" (i.e., the live server), but instead teaches its staging area as being implemented on the development server 130. That is, as described above, *Nakano* does not teach creating a staging folder on its live server 170, but instead merely proposes to create a staging area on a separate development server 130. Accordingly, *Nakano* fails to anticipate claim 1 for at least this further reason.

Additionally, *Nakano* fails to teach a staging folder that does not default to a directory listing of said file system when accessed. Instead, *Nakano* teaches that its staging space is a branch that would appear to be listed as part of the directory listing of the development server's file system. In asserting that *Nakano* teaches this element of claim 1, the Final Office Action (at page 2 thereof) merely cites to col. 2, lines 48-52 of *Nakano*, which provides:

> A staging area is a read-only file system that supports select versioning operations. Various users of work areas can integrate their work by submitting the contents of their work areas to the staging area. In the staging area, developers can compare their work and see how their changes fit together.

Clearly, the above portion of *Nakano* fails to provide any teaching whatsoever of its staging area not defaulting to inclusion in a directory listing of the file system. Indeed, *Nakano* mentions throughout its disclosure that the staging area (as well as the other "areas") of its file system may be included as branches of a directory tree, *see e.g.*, col. 2, lines 58-64; col. 5, lines 39-56; and col. 6, lines 30-32, 45-47, and 59-64. Thus, it appears that nothing in *Nakano* would prevent its staging area from defaulting to a directory listing of the file system. Instead, any such directory listing of the file system in *Nakano* could include its staging area(s). Thus, *Nakano* fails to anticipate claim 1 for at least this further reason.

Finally, *Nakano* fails to teach "inserting a randomized string into a name of said file assets to be staged". In asserting that *Nakano* teaches this element of claim 1, the Final Office Action (at page 2 thereof) merely cites to col. 8, lines 18-29 and 52-62 of *Nakano*, which provide:

> Each work area, staging area, and edition area has two unique identifiers, one of which is referred to in this application as a "generation ID," and the other of which is referred to as an "object ID." The object ID identifies the object that represents the area, and, once an object ID is assigned to an object, that object ID is not changed. Each area is also identified by a unique generation ID, which indicates how an area is related to other areas. The generation ID for a particular area can be changed, as will be discussed below (e.g., when a staging area is published into an edition). The generation ID is placed in the generation ID field. The object ID is placed in the object ID field. (Lines 18-29).

> FIG. 10 illustrates the method for deriving a generation ID, which is to be assigned to a new area, from a parent generation ID, where the parent generation ID is assigned to the direct parent of the new area. A unique number is obtained 1000 using a conventional algorithm for sequentially (e.g., 1, 2, 3, 4) or randomly generating unique numbers. The set or sequence of unique numbers associated with the parent generation ID is then retrieved 1010. Subsequently, a set or sequence of numbers that is the concatenation of the parent generation ID and the just issued unique number is created 1020. (Lines 52-62).

While the above portion of *Nakano* mentions a two identifiers – an object ID and a generation ID – *Nakano* fails to teach inserting any random string into a name of the file assets to be staged. For instance, *Nakano* teaches that the object ID and generation ID are separate fields for an area from the area's name. For example, at col. 8, lines 6-10, *Nakano* explains:

> Each separate work area, staging area, and edition area is created by creating an object that represents the area and that has a name field, an object ID field, a generation ID field, a directory field, and a branch field. The name field includes the name of the work area.

Thus, *Nakano* fails to teach this further element of claim 1 of "inserting a randomized string into a name of said file assets to be staged". Accordingly, *Nakano* fails to anticipate claim 1 for at least this further reason.

In view of the above, *Nakano* fails to teach all elements of claim 1, and therefore the rejection of claim 1 should be withdrawn.

Independent Claim 8

Independent claim 8 recites:

> A computer program product having a computer readable medium with computer program logic recorded thereon for facilitating staging file assets, wherein said computer program logic comprises code that when executed by a computer causes the computer to perform a method comprising:
> detecting, in a file system of a Web server, an index of said Web server;
> generating, in said file system of said Web server, a staging folder;
> storing, in said staging folder, said file assets to be staged according to names that include a random string, wherein said file assets are served by said Web server to users accessing said staging folder; and

inhibiting listing of said staging folder in a default directory listing of said file system by said Web server. (Emphasis added).

Nakano fails to teach at least the above-emphasized elements of claim 8. First, as discussed above with claim 1, *Nakano* fails to teach storing file assets to be staged according to names that include a random string. Additionally, as discussed above with claim 1, *Nakano* fails to teach "inhibiting listing of said staging folder in a default directory listing of said file system by said Web server". Nothing in *Nakano* inhibits the listing of its stage area in a default directory listing of the file system.

For at least the above reasons, *Nakano* fails to teach all elements of claim 8, and therefore the rejection of claim 8 should be withdrawn.

Independent Claim 14

Independent claim 14 recites:

A method for reviewing proposed file content on a live Web server comprising:
scanning said live Web server for an index file;
opening a review folder in a file system of said live Web server;
creating a blank index on said review folder, wherein said blank index is named according to a name of said index file, wherein said blank index inhibits listing by said Web server of said review folder in a default directory listing of said file system; and
storing said proposed file content in said review folder, wherein a randomized string is included in a file name representing said proposed file content, and wherein said proposed file content is served by said Web server to users accessing said file name. (Emphasis added).

Nakano fails to teach at least the above-emphasized elements of claim 14. First, as discussed above with claim 1, *Nakano* fails to teach a method for reviewing proposed file content on a "live Web server" (such as its live server 170), but instead discloses a staging area on a separate development server 130.

Additionally, *Nakano* fails to teach creating a blank index that inhibits listing of a review folder by such live Web server in a default directory listing of the file system. *Nakano* does not

teach creating any such blank index (nor any other mechanism) that inhibits listing of its staging area(s) in a default directory listing of the file system.

Finally, as discussed above with claim 1, *Nakano* does not teach inclusion of a randomized string in a file name representing the proposed file content. While *Nakano* mentions certain ID fields that are created for an area, it does not teach that these or any other randomized strings are included in a file name.

For at least the above reasons, *Nakano* fails to teach all elements of claim 14, and therefore the rejection of claim 14 should be withdrawn.

Dependent Claims

Each of dependent claims 2-7, 9-13, and 15-20 depends, either directly or indirectly, from one of independent claims 1, 8, and 14 (and thus inherits all limitations of its respective independent claim). In view of the above, Applicant respectfully submits that independent claims 1, 8, and 14 are of patentable merit. It is respectfully submitted that dependent claims 2-7, 9-13, and 15-20 are allowable at least because of their dependency from their respective independent claims for the reasons discussed above.

III. Conclusion

In view of the above, applicant believes the pending application is in condition for allowance.

Applicant believes no fee is due with this response. However, if a fee is due, please charge our Deposit Account No. 06-2380, under Order No. M060 from which the undersigned is authorized to draw.

Dated: March 10, 2008

Respectfully submitted,

I hereby certify that this paper (along with any paper referred to as being attached or enclosed) is being transmitted via the Office electronic filing system in accordance with § 1.6(a)(4).

Dated: March 10, 2008

Signature: Donna Forbit
(Donna Forbit)

By _____
Jody C. Bishop
Registration No.: 44,034
FULBRIGHT & JAWORSKI L.L.P.
2200 Ross Avenue, Suite 2800
Dallas, Texas 75201-2784
(214) 855-8007
(214) 855-8200 (Fax)
Attorney for Applicant